

CYCLIC COORDINATE UPDATE ALGORITHMS FOR FIXED-POINT PROBLEMS: ANALYSIS AND APPLICATIONS

YAT TIN CHOW*, TIANYU WU*, AND WOTAO YIN*

Abstract. Many problems reduce to the fixed-point problem of solving $x = T(x)$. To this problem, we apply the coordinate-update algorithms, which update only one or a few components of x at each step. When each update is cheap, these algorithms are faster than the full fixed-point iteration (which updates all the components).

In this paper, we focus on the coordinate-update algorithms based on the cyclic selection rules, where the ordering of coordinates in each cycle is arbitrary. These algorithms are fast, but their convergence is unknown in the fixed-point setting.

When T is a nonexpansive operator and has a fixed point, we show that the sequence of coordinate-update iterates converges to a fixed point under proper step sizes. This result applies to the primal-dual coordinate-update algorithms, which have applications to optimization problems with nonseparable nonsmooth objectives, as well as global linear constraints.

Numerically, we apply coordinate-update algorithms with the cyclic, shuffled cyclic, and random selection rules to ℓ_1 robust least squares, a CT image reconstruction problem, as well as nonnegative matrix factorization. They converge much faster than the standard fixed-point iteration. Among the three rules, cyclic and shuffled cyclic rules are overall faster than the random rule.

Key words. coordinate update, cyclic, shuffled cyclic, fixed point, nonexpansive operator, robust least squares, image reconstruction, nonnegative matrix factorization

AMS subject classifications. 90C06, 90C25, 65K05

1. Introduction. We recently witnessed a strong demand for fast numerical solutions for large-scale problems. Numerical methods of small memory footprints become very popular. Among them *coordinate update algorithms* (e.g., [3, 4, 24, 36, 26, 32, 41, 7, 29, 28]) are found to be very useful. They reformulate a problem as a fixed-point problem and decompose it further into simple subproblems, each updating one, or a small block of, variables while fixing others. They are popular numerical choices to problems with the *coordinate-friendly structure*, namely, there are means to update a component, or a small block of components, of its variable much cheaper than updating all components of the variable. A variety of problems, including second order cone programming, variational image processing, support vector machines, empirical risk minimization, portfolio optimization, distributed computing, and nonnegative matrix factorization, have this structure [28].

A class of coordinate update algorithms is the *coordinate descent algorithms* for optimization, where the objective function, or a surrogate function, is reduced at each iteration; e.g., see [2, 32]. In randomized coordinate descent algorithms such as [26, 23, 33], it is the the (conditional) expectation of the objective function that descends iteratively. Coordinate descent algorithms are efficient at solving problems with Lipschitz differentiable objective functions, as well as those with *separable* non-differentiable functions and constraints. See the recent survey papers [39, 34, 43].

¹Department of Mathematics, University of California, Los Angeles, CA 90095, USA, yatchow@ucla.edu, wutyu11@math.ucla.edu. This research is supported by the NSF grant ECCS-1462397.

On the other hand, there exist non-separable, non-differentiable examples [38] [34, Section 2.2.1] on which coordinate descent algorithms get stuck at a non-stationary point. Also, it is difficult for coordinate descent algorithms to directly handle problems with global constraints since they involve most or all components of variables. Such examples are found in problems such as ℓ_1 -robust least squares, total variation image processing, and the extended monotropic program. They can be solved by the recent primal-dual coordinate-update algorithms [30, 7, 12] [28, Section 4]. Primal-dual coordinate-update algorithms do not necessarily produce a sequence of monotonic objective values, so it is challenging to analyze their convergence in the coordinate descent framework. However, they fit the setting of this paper as fixed-point algorithms. In fact, the primal-dual coordinate update algorithms under various cyclic selection rules are new to the best of our knowledge.

In addition to optimization problems, fixed-point iterations arise in variational inequalities [40], inverse problems [17], equilibrium analysis [19, 9] and control theory [27], where the fixed-point operators are nonexpansive.

This paper focuses on the fixed-point problem with a nonexpansive operator and studies the convergence of its coordinate-update algorithms. Despite the rich literature on the fixed-point problem, our understanding to its coordinate-update algorithms is very limited. In the literature, there are two classes of convergence analysis. They use different metrics for the distance between the current iterate and the fixed point (or one in the set of fixed points). The first class uses the (weighted) ℓ_∞ distance [3, 4]. Their applications include linear and nonlinear systems with dominant diagonals, as well as certain optimization problems with a smooth objective and simple constraints. The second class uses the ℓ_2 distance and has potentially more applications. However, convergence results are limited to the random selection of coordinates [7, 29][28, Appendix D]. However, we prefer the cyclic selection over the random selection. But the convergence of the former has not been studied under the ℓ_2 distance.

Let us briefly compare the random and cyclic selection rules. The random selection is easier to analyze since taking expectation reduces it to the analysis of the standard (full update) fixed-point iteration. However, we found that cyclic selection took fewer iterations to converge than random selection in all of our numerical tests (presented in Section 4 below). The advantages of cyclic selection has also been observed with the coordinate descent algorithms in statistics and optimization [13, 14]. Cyclic selection is also more cache efficient under the contemporary computer architecture. Data of adjacent coordinates are typically stored at consecutive memory locations, so they are transmitted into *caches* in batches. Cyclic selection accesses these data sequentially, likely encountering *cache hits*. Random selection, on the contrary, accesses data randomly, likely encountering *cache misses*. Hence, random selection is less cache efficient. Moreover, random selection also requires pseudo-random number generation, which can take more time than the coordinate updates if the underlying problem is coordinate-friendly with very cheap updates.

Our exposition in this paper was motivated by the observations that the cyclic coordinate update algorithms have excellent numerical performance for many fixed-point problems

yet their convergence has not been understood yet.

1.1. Problem formulation. Given an operator $S : \mathcal{H} \rightarrow \mathcal{H}$, our problem is to find $x \in \mathcal{H}$ such that

$$(1.1) \quad S(x) = 0.$$

This problem is equivalent to finding a fixed point to the operator $T = I - S$. (In the rest of this paper we choose to use S instead of T because it is overall more convenient to present our results in S .) Our assumption to this problem is that the operator $I - S$ has a fixed point (possibly not unique) and it is nonexpansive, that is,

$$(1.2) \quad \|(I - S)x - (I - S)y\| \leq \|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

The condition (1.2) is equivalent to S being $(1/2)$ -cocoercive, namely,

$$(1.3) \quad \frac{1}{2}\|Sx - Sy\|^2 \leq \langle Sx - Sy, x - y \rangle, \quad \forall x, y \in \mathcal{H}.$$

1.2. Contributions. This paper proposes the standard and shuffled cyclic coordinate update algorithms to solve (1.1) on a Hilbert space \mathcal{H} . The contributions of this paper include:

1. We take the standard setting that the operator $I - S$ is nonexpansive and has a fixed point. We show that the cyclic coordinate-update algorithms converge to a fixed point, as long as proper step sizes are chosen. This result holds for all orderings of the coordinates as long as each is updated every cycle. For example, one can shuffle the coordinates or rank the coordinates based on their coordinate-wise Lipschitz constants.

We show in Theorem 3.3 that a sequence of $O(\frac{1}{\sqrt{k}})$ step sizes ensures convergence. If the operator S is further quasi- μ -strongly monotone (c.f. Assumption 2 below), then a fixed step size is sufficient for convergence (Theorem 3.5). However, the step size is still inversely proportional to the total number of coordinates. Nonetheless, the unit, fixed step size worked for all of our tested problems.

Our proofs are based on comparing the operator of cyclic coordinate update to the standard full update operator and bounding the extra error due to the cyclic updates. This approach is significantly different from those for cyclic coordinate descent [2, 32] (based on function value descent) and fixed-point random coordinate update [7, 29] (taking expectations and using super-martingale convergence).

2. Based on the algorithms in part 1, we propose specific coordinate-update algorithms for two problems: ℓ_1 -robust least squares and total-variation based computer tomograph (CT) reconstruction. We also tested an existing algorithm for nonnegative matrix factorization. We briefly explain how to apply coordinate updates by exploiting their coordinate-friendly structures.

For all three problems, we found that cyclic coordinate updates, either deterministic or shuffled, took fewer epochs¹ to reach the same accuracy than randomized coordinate update, and these coordinate updates took fewer epochs than (full update) fixed-point iteration.

Our numerical results were obtained with the optimal fixed step size parameters for all algorithms. We conjecture that fixed step sizes sufficiently ensure the convergence of our algorithms when the fixed-point problem is derived from an optimization problem via the primal-dual method.

1.3. Algorithm. In this paper our variable has m blocks, $x_i \in \mathcal{H}_i$, $i = 1, \dots, m$, where \mathcal{H}_i is a Hilbert space. Their Cartesian product is $\mathcal{H} := \mathcal{H}_1 \times \dots \times \mathcal{H}_m$.

For each coordinate $i = 1, \dots, m$, we define the i th coordinate operator $S_i : \mathcal{H} \rightarrow \mathcal{H}$ such that

$$S_i(x) = (0, \dots, (S(x))_i, \dots, 0), \quad \forall x \in \mathcal{H}.$$

We establish the convergence of the (shuffled) cyclic coordinate-update algorithm (Algorithm 1) and numerically demonstrate its efficiency.

Algorithm 1: Cyclic Coordinate Update

```

input:  $x^0 \in \mathcal{H}$ 
1 for  $k = 1, 2, \dots$ , do
2   set  $(i_1, i_2, \dots, i_m)$  as a permutation of  $(1, 2, \dots, m)$ ;
3   (either no permutation, random shuffling, or greedy ordering);
4   initialize  $y^0 \leftarrow x^{k-1}$ ;
5   for  $j = 1, \dots, m$  do
6     set  $y^j \leftarrow y^{j-1} - \alpha_k S_{i_j}(y^{j-1})$ ;
7   set  $x^k \leftarrow y^m$ ;
```

In Algorithm 1, each k is an outer loop or an epoch, and each j is an inner loop. Line 3 sets an initial point for the inner loop, then Line 4–5 update each of the m coordinates once, and finally Line 6 finishes the inner loop by updating x^k . Note that each inner iteration step only updates the i_j th coordinate:

$$(y^j)_i = \begin{cases} (y^{j-1})_i - \alpha_k (S(y^{j-1}))_i, & \text{if } i = i_j \\ (y^{j-1})_i, & \text{otherwise,} \end{cases} \quad \text{for } i = 1, \dots, m.$$

The order of the m coordinates is specified at the beginning of each epoch. Typical choices include the ordering $(1, 2, \dots, m)$, its random shuffling, and a greedy ordering, e.g. in the descending order of their coordinate-wise Lipschitz constants. We can also *shuffle only in the first epoch* and then fix the ordering for all remaining epochs. All these ordering rules are numerically efficient, and our analysis applies to all of them.

¹An epoch consists of m coordinate updates, where m is the total number of coordinates.

The non-shuffling cyclic ordering is the easiest to code and has the cheapest per-iteration cost because shuffling requires pseudo-random number generation, which can be time consuming, especially when each inner iteration step is simple. Shuffling, however, avoids the worst ordering and, depending on the application, can accelerate the convergence.

1.4. Organization. The rest of this paper is organized as follows. Section 2 briefly reviews fixed-point iteration of nonexpansive operators and its application. Section 3 presents our theoretical analysis. Numerical results and further applications are presented in Section 4. Finally, Section 5 concludes this paper.

2. Background. A traditional algorithm for Problem (1.1) is the Krasnosel'skii-Mann (KM) iteration [20, 25]:

$$(2.1) \quad x^{k+1} = x^k - \eta_k Sx^k,$$

where η_k is a step size parameter.

KM iteration (2.1) has many special cases such as gradient descent, proximal-point, prox-gradient, as well as many operator-splitting algorithms including forward-backward splitting, Peaceman-Rachford splitting, Douglas-Rachford splitting [22], the alternating direction of multipliers (ADMM) [16, 15], three-operator splitting [10], primal-dual splitting [8, 37]. Many of their variations are special examples of KM iteration, too. For each of these algorithms, one can recover an operator S such that $(I - S)$ is nonexpansive though this recovery is not obvious. Therefore, the convergence (in sequence) follows from the analysis of the KM iteration.

2.1. Notation and preliminaries. The solution to our problem (1.1) is the zero set of S :

$$\text{zer}(S) := \{x \in \mathcal{H} : 0 = Sx\}.$$

The minimal assumption that we make to the problem is given below.

ASSUMPTION 1. *$\text{zer}(S)$ is nonempty. $(I - S)$ is nonexpansive.*

As already mentioned, $(I - S)$ is nonexpansive, i.e., obeying (1.2), if and only if S is $(1/2)$ -cocoercive, that is, S obeys inequality (1.3). That inequality implies that S is 2-Lipschitz:

$$(2.2) \quad \|Sx - Sy\| \leq 2\|x - y\|, \quad \forall x, y \in \mathcal{H}.$$

Since $\|S_i x - S_i y\| \leq \|Sx - Sy\|$, each S_i is 2-Lipschitz, too. However, the Lipschitz constants for S_i can be (much) smaller than 2, permitting theoretically larger step sizes. Hence, **we let L_i be the Lipschitz constant of S_i , i.e.,**

$$\|S_i x - S_i y\| \leq L_i \|x - y\|, \quad \forall x, y \in \mathcal{H},$$

and, for simplicity, let

$$(2.3) \quad L := \max_i L_i \leq 2.$$

Properly replacing L by different L_i in our analysis below may improve our results, but we prefer simplicity over better bounds.

Next, we switch our focus to KM iteration and Algorithm 1. Without loss of generality, we fix the coordinate updating order as 1 through m .

DEFINITION 2.1 (the full and cyclic coordinate update operators).

For $0 < \alpha < 1$, we define the full update operator and cyclic coordinate update operator, respectively, as

$$(2.4) \quad T^\alpha := I - \alpha S,$$

$$(2.5) \quad E^\alpha := (I - \alpha S_m)(I - \alpha S_{m-1}) \cdots (I - \alpha S_1).$$

When $(I - S)$ is nonexpansive, the operator T^α has the following properties [1, Prop. 4.25]:

$$(2.6) \quad \|T^\alpha x - T^\alpha y\|^2 \leq \|x - y\|^2 - \alpha(1 - \alpha)\|Sx - Sy\|^2, \quad \forall x, y \in \mathcal{H}.$$

Such an operator is called an *averaged operator* since $T^\alpha = (1 - \alpha)I + \alpha(I - S)$.

For any $x^* \in \text{zer}(S)$, substituting $y = x^*$ in (2.6) and noticing $T^\alpha x^* = x^* - \alpha Sx^* = x^*$ yield the quasi²-contractive property:

$$(2.7) \quad \|T^\alpha x - x^*\|^2 \leq \|x - x^*\|^2 - \alpha(1 - \alpha)\|Sx\|^2, \quad \forall x \in \mathcal{H},$$

which is a key property for the convergence of the KM iteration $x^{k+1} = T^\alpha x^k$.

The operator E^α characterizes one epoch of Algorithm 1 under the cyclic ordering $1, 2, \dots, m$. Indeed, the iterates x^k of Algorithm 1 satisfy

$$x^{k+1} = E^{\alpha_k}(x^k).$$

The following proposition follows directly from (2.2) (2.3) and (2.5).

PROPOSITION 2.2. *The operator E^{α_k} is $(1 + \alpha_k L)^m$ -Lipschitz.*

We will set α_k so that an inequality similar to (2.7) holds for E^{α_k} and x^k (weakly) converges to a point in $\text{zer}(S)$.

3. Convergence results. Our analysis is based on comparing the operator E^α with the operator T^α . To simplify notation, let us define the operator

$$(3.1) \quad R := \frac{1}{\alpha}(T^\alpha - E^\alpha).$$

²The modifier *quasi* is used if the property involves the solution x^* .

We have

$$\begin{aligned}
R &= \frac{1}{\alpha}(T^\alpha - E^\alpha) \\
&= \frac{1}{\alpha}[(I - \alpha S) - (I - \alpha S_m)(I - \alpha S_{m-1}) \cdots (I - \alpha S_1)] \\
&= \sum_{i=2}^m \left(S_i(I - \alpha S_{i-1})(I - \alpha S_{i-2}) \cdots (I - \alpha S_1) - S_i \right),
\end{aligned}$$

and hence we get that

$$\|Rx\|^2 = \sum_{i=2}^m \|S_i x - S_i(I - \alpha S_{i-1})(I - \alpha S_{i-2}) \cdots (I - \alpha S_1)x\|^2.$$

In fact we have the following estimate for R .

LEMMA 3.1. *The operator R satisfies the estimate*

$$(3.2) \quad \|Rx\| \leq \frac{\alpha L m}{\sqrt{2}}(1 + \alpha L)^m \|Sx\|.$$

Proof. Let us consider for each i ,

$$\begin{aligned}
\Delta_i &:= \|S_i x - S_i(I - \alpha S_{i-1})(I - \alpha S_{i-2}) \cdots (I - \alpha S_1)x\| \\
&\leq L\|x - (I - \alpha S_{i-1})(I - \alpha S_{i-2}) \cdots (I - \alpha S_1)x\|.
\end{aligned}$$

The triangle inequality yields

$$\begin{aligned}
\Delta_i &\leq L\|x - (I - \alpha S_{i-1})x\| + L\|(I - \alpha S_{i-1})x - (I - \alpha S_{i-1})(I - \alpha S_{i-2})x\| + \cdots \\
&\quad + L\|(I - \alpha S_{i-1}) \cdots (I - \alpha S_2)x - (I - \alpha S_{i-1}) \cdots (I - \alpha S_1)x\|.
\end{aligned}$$

Applying Proposition 2.2, we obtain

$$\begin{aligned}
\Delta_i &\leq L\|x - (I - \alpha S_{i-1})x\| + L(1 + \alpha L)\|x - (I - \alpha S_{i-2})x\| \\
&\quad + L(1 + \alpha L)^2\|x - (I - \alpha S_{i-3})x\| + \cdots + L(1 + \alpha L)^{i-2}\|x - (I - \alpha S_1)x\| \\
&\leq \alpha L\|S_{i-1}x\| + \alpha L(1 + \alpha L)\|S_{i-2}x\| \\
&\quad + \alpha L(1 + \alpha L)^2\|S_{i-3}x\| + \cdots + \alpha L(1 + \alpha L)^{i-2}\|S_1x\| \\
&\leq \alpha L(1 + \alpha L)^m \sum_{j=1}^{i-1} \|S_j x\|
\end{aligned}$$

By the Cauchy-Schwarz inequality,

$$\Delta_i \leq \alpha L(1 + \alpha L)^m \sqrt{i-1} \|Sx\|.$$

Finally, combining the above inequalities yields

$$\|Rx\|^2 = \sum_{i=2}^m \Delta_i^2 \leq \sum_{i=2}^m (i-1)\alpha^2 L^2 (1+\alpha L)^{2m} \|Sx\|^2 \leq \frac{\alpha^2 L^2 m^2}{2} (1+\alpha L)^{2m} \|Sx\|^2.$$

□

Pick an arbitrary $x^* \in \text{zer}(S)$. To use the property (2.7) of T^α , we expand $\|E^\alpha x - x^*\|^2$ by $E^\alpha = T^\alpha - \alpha R$ as follows:

$$\begin{aligned} \|E^\alpha x - x^*\|^2 &= \|T^\alpha x - \alpha R x - x^*\|^2 \\ (3.3) \quad &= \|T^\alpha x - x^*\|^2 - 2\alpha \langle T^\alpha x - x^*, Rx \rangle + \alpha^2 \|Rx\|^2. \end{aligned}$$

By Young's inequality, the cross term in (3.3) satisfies

$$(3.4) \quad -2\alpha \langle T^\alpha x - x^*, Rx \rangle \leq \alpha \eta \|T^\alpha x - x^*\|^2 + \alpha \eta^{-1} \|Rx\|^2$$

for any $\eta > 0$, which we will set later. Substituting (3.4) into (3.3) and then applying Lemma 3.1 yield

$$\begin{aligned} \|E^\alpha x - x^*\|^2 &\leq (1 + \alpha \eta) \|T^\alpha x - x^*\|^2 + \alpha (\eta^{-1} + \alpha) \|Rx\|^2 \\ (3.5) \quad &\leq (1 + \alpha \eta) \left(\|T^\alpha x - x^*\|^2 + \eta^{-1} \alpha^3 L^2 \frac{m^2}{2} (1 + \alpha L)^{2m} \|Sx\|^2 \right). \end{aligned}$$

Substituting $x = x^k$ and $\alpha = \alpha_k$ and using (2.7) yield

$$(3.6) \quad \|E^{\alpha_k} x^k - x^*\|^2 \leq (1 + \alpha_k \eta) \left(\|x^k - x^*\|^2 - \left(\alpha_k (1 - \alpha_k) - \alpha_k^3 L^2 \frac{m^2 (1 + \alpha_k L)^{2m}}{2\eta} \right) \|Sx^k\|^2 \right)$$

for all k .

The existence of the extra coefficient $\alpha_k \eta$ in (3.6) invalidates the traditional analysis. To ensure convergence, we take two approaches: using slowly decreasing step sizes in Subsection 3.1 and making stronger assumptions on S in Subsection 3.2.

3.1. Slowly decreasing step sizes. Our convergence will reduce to the analysis of some simple scalar sequences as follows.

LEMMA 3.2. *Consider the sequences*

$$(a_k)_{k \geq 0}, (b_k)_{k \geq 0}, (\xi_k)_{k \geq 0} \subset \{x \in \mathbb{R} : x \geq 0\},$$

where $\sum_{k \geq 0} \xi_k < \infty$ and

$$(3.7) \quad a_{k+1} \leq (1 + \xi_k)(a_k - b_k).$$

Then, (i) $(a_k)_{k \geq 0}$ is bounded, (ii) there exists $a^* \in \mathbb{R}_+$ such that $\lim_k a_k = a^* \in \mathbb{R}_+$, and (iii) $\sum_{k \geq 0} b_k < \infty$.

In the simplified case $\xi_k \equiv 0$, the results hold trivially. Indeed, since $a_k \geq 0$ and

$a_{k+1} \leq a_k$ by (3.7), there exists $a^* \in \mathbb{R}_+$ such that $a_k \rightarrow a^*$, and the telescoping sum of (3.7) yields $\sum_{k \geq 0} b_k < \infty$. Lemma 3.2 claims that these results still hold under the multiplicative errors ξ_k that are summable.

Proof. Expanding (3.7) yields

$$(3.8) \quad \begin{aligned} a_{k+1} &\leq (1 + \xi_k)(1 + \xi_{k-1})a_{k-1} - ((1 + \xi_k)b_k + (1 + \xi_k)(1 + \xi_{k-1})b_{k-1}) \\ &\leq \dots \leq (\Pi_{j=0}^k (1 + \xi_j))a_0 - \sum_{j=0}^k ((\Pi_{i=j}^k (1 + \xi_i))b_j). \end{aligned}$$

Let $\xi := \Pi_{j=0}^k (1 + \xi_j)$. Noticing $(1 + \xi_j) \leq e^{\xi_j}$ for $\xi_j \geq 0$ and using $\sum_{k \geq 0} \xi_k < \infty$ gives us $\xi < \infty$. Applying the inequality $\sum_{j=0}^k ((\Pi_{i=j}^k (1 + \xi_i))b_j) \geq \sum_{j=0}^k b_j$ to (3.8), we obtain

$$a_{k+1} \leq \xi a_0 - \sum_{j=0}^k ((\Pi_{i=j}^k (1 + \xi_i))b_j) \leq \xi a_0 - \sum_{j=0}^k b_j,$$

which means that $a_k \in [0, \xi a_0]$ for all $k \geq 0$ and $\sum_{j \geq 0} b_j \leq \xi a_0 < \infty$. We have proved Parts (i) and (iii).

Let $d_0 := a_0$ and $d_k := a_k - a_{k-1}, k = 1, 2, \dots$. Below we show that $\sum_{k=0}^{\infty} |d_k| < \infty$. To this end, let

$$d_k^+ := \max\{d_k, 0\} \text{ and } d_k^- := -\min\{d_k, 0\}, \forall k,$$

which yield $d_k^+ + d_k^- = |d_k|$ and $d_k^+ - d_k^- = d_k$. By $a_{k+1} \leq (1 + \xi_k)(a_k - b_k)$, $a_k \leq \xi a_0$, and $b_k \geq 0$, we have $a_{k+1} - a_k \leq \xi_k \xi a_0$, which means $d_{k+1}^+ \leq \xi_k \xi a_0, \forall k$ and thus

$$\sum_{k=0}^{\infty} d_k^+ \leq a_0 + \xi a_0 \sum_{k=0}^{\infty} \xi_k < \infty.$$

From $0 \leq a_k = \sum_{i=0}^k (d_i^+ - d_i^-)$, $\forall k$, we obtain $\sum_{i=0}^k d_k^- \leq \sum_{i=0}^k d_k^+$ and thus $\sum_{i=0}^{\infty} d_k^- < \infty$. Finally,

$$\sum_{k=0}^{\infty} |d_k| = \sum_{k=0}^{\infty} d_k^+ + \sum_{k=0}^{\infty} d_k^- < \infty.$$

Hence, $\{a_k\}_{k \geq 0}$ is a Cauchy sequence, and Part (ii) holds. \square

By applying Lemma 3.2, we can establish the following convergence result.

THEOREM 3.3. *Let Algorithm 1 use the step sizes*

$$(3.9) \quad \alpha_k = \frac{1}{k^{1/2}}.$$

Under Assumption 1, x^k (weakly) converges to x^ for some $x^* \in \text{zer}(S)$.*

Proof. We choose $\eta = m^2 L^2 \alpha_k^2 (1 + \alpha_k L)^{2m}$ in (3.6) to get

$$\|x^{k+1} - x^*\|^2 \leq (1 + m^2 L^2 \alpha_k^3 (1 + \alpha_k L)^{2m}) \left(\|x^k - x^*\|^2 - \alpha_k \left(\frac{1}{2} - \alpha_k \right) \|Sx^k\|^2 \right).$$

With (3.9) and $(1 + \alpha_k L)^{2m} \leq (1 + L)^{2m}$, we have $\sum_k m^2 L^2 \alpha_k^3 (1 + \alpha_k L)^{2m} < \infty$, and $\alpha_k (\frac{1}{2} - \alpha_k) \geq 0$ for $k \geq 4$. Since E^{α_k} is $(1 + \alpha_k L)^m$ -Lipschitz and $\alpha_k \leq 1$, $\|x^4 - x^*\| \leq (1 + L)^4 \|x^0 - x^*\|$.

By Lemma 3.2 (applied to $k \geq 4$), $\|x^{k+1} - x^*\|$ converges to some $c \geq 0$, and thus $(x^k)_{k \geq 0}$ is bounded and has a weak cluster point \bar{x} , and

$$(3.10) \quad \sum_{k \geq 4} \alpha_k \left(\frac{1}{2} - \alpha_k \right) \|Sx^k\|^2 < \infty.$$

Since $\frac{1}{4\sqrt{k}} \leq \alpha_k (\frac{1}{2} - \alpha_k) \leq \frac{1}{2\sqrt{k}}$ for $k \geq 16$, we obtain from (3.10):

$$(3.11) \quad \sum_{k \geq 4} \frac{1}{\sqrt{k}} \|Sx^k\|^2 < \infty$$

Next, we take a few steps to prove $\lim_{k \rightarrow \infty} \|Sx^k\| = 0$ using the summability of $\frac{1}{\sqrt{k}} \|Sx^k\|^2$ and the Lipschitz continuity of S .

In the following we show that $|\|Sx^{k+1}\|^2 - \|Sx^k\|^2| \leq O(k^{-1/2})$.

By Lemma 3.2, there exists $A \in \mathbb{R}$ such that $\|x_k - x^*\| \leq A$. We have

$$\|x^{k+1} - x^k\| = \|E^{\alpha_k} x^k - x^k\| = \|T^{\alpha_k} x^k - \alpha_k R x^k - x^k\| = \|-\alpha_k S x^k - \alpha_k R x^k\|.$$

By $Sx^* = 0$, the triangle inequality, and (3.2),

$$\begin{aligned} \|x^{k+1} - x^k\| &\leq \alpha_k \|Sx^k - Sx^*\| + \alpha_k \|R x^k\| \\ &\leq \left(\alpha_k + \frac{\alpha_k^2 L m}{\sqrt{2}} (1 + \alpha_k L)^m \right) \|Sx^k - Sx^*\|. \end{aligned}$$

Because S is 2-Lipschitz, $\alpha_k \leq 1$, and $\|x_k - x^*\| \leq A$,

$$\begin{aligned} \|x^{k+1} - x^k\| &\leq 2\alpha_k \left(1 + \frac{\alpha_k L m}{\sqrt{2}} (1 + \alpha_k L)^m \right) \|x^k - x^*\| \\ &\leq 2\alpha_k \left(1 + \frac{L m}{\sqrt{2}} (1 + L)^m \right) A \\ &\leq 2 \left(1 + \frac{L m}{\sqrt{2}} (1 + L)^m \right) A \cdot k^{-\frac{1}{2}}. \end{aligned}$$

Then we have

$$\begin{aligned}
|\|Sx^{k+1}\|^2 - \|Sx^k\|^2| &= |\|Sx^{k+1}\| - \|Sx^k\|| (\|Sx^{k+1}\| + \|Sx^k\|) \\
&\leq \|Sx^{k+1} - Sx^k\| (\|Sx^{k+1} - Sx^*\| + \|Sx^k - Sx^*\|) \\
&\leq 2\|x^{k+1} - x^k\| (2\|x^{k+1} - x^*\| + 2\|x^k - x^*\|) \\
&\leq \underbrace{2 \cdot 2 \left(1 + \frac{Lm}{\sqrt{2}}(1+L)^m\right) A \cdot 4A \cdot k^{-\frac{1}{2}}}_{=:B} = Bk^{-\frac{1}{2}}.
\end{aligned}$$

Now we show $\lim_{k \rightarrow \infty} \|Sx^k\|^2 = 0$ by contradiction. Suppose $C := \limsup_{k \rightarrow \infty} \|Sx^k\|^2 > 0$. For each $N > 0$, there exists $n \geq N$ such that $F := \|Sx^n\|^2 \geq \frac{C}{2}$. Since $\|Sx^{k+1}\|^2 - \|Sx^k\|^2 \geq -Bk^{-\frac{1}{2}}$, we have $\|Sx^{n+i}\|^2 \geq \frac{F}{2}$ for all $0 \leq i \leq \lfloor n' \rfloor$, where $n' = \frac{F\sqrt{n}}{2B}$. Then we have

$$\begin{aligned}
\sum_{i=n}^{n+\lfloor n' \rfloor} i^{-1/2} \|Sx^i\|^2 &\geq \frac{F}{2} \sum_{i=n}^{n+\lfloor n' \rfloor} i^{-1/2} \geq \frac{F}{2} \int_n^{n+n'} x^{-1/2} dx \\
&\geq F(\sqrt{n+n'} - \sqrt{n}) = \frac{F^2\sqrt{n}}{2B(\sqrt{n+n'} + \sqrt{n})}
\end{aligned}$$

When n is large enough, we have $n' \leq 3n$, so the last term is at least $\frac{F^2}{6B}$. Therefore, by Cauchy's criterion, the sequence $\frac{1}{\sqrt{k}} \|Sx^k\|^2$ is not summable, which contradicts (3.11). Hence, we have $\lim_{k \rightarrow \infty} \|Sx^k\|^2 = 0$.

Now we adapt the convergence proof of KM iteration [20] to our setting.

Recall the demicloseness principle [1]: if T is nonexpansive, $z^j \rightharpoonup \bar{z}$, and $\lim \|(I - T)z^j\| = 0$, then $\bar{z} = T(\bar{z})$. Applying this principle to $T = I - S$ and the subsequence of $(x^k)_{k \geq 0}$ that weakly converges to \bar{x} , we obtain $\bar{x} \in \text{zer}(S)$.

Following the proof in [1], we show that any weak cluster point \bar{y} of $(x^k)_{k \geq 0}$ must equal \bar{x} . The demicloseness principle again yields $\bar{y} \in \text{zer}(S)$. Substituting $x^* = \bar{x}$ and then $x^* = \bar{y}$ and following the argument above yield the limits $\lim_k \|x^{k+1} - \bar{x}\| = c_x$ and $\lim_k \|x^{k+1} - \bar{y}\| = c_y$. Algebraically,

$$(3.12) \quad 2\langle x^k, \bar{x} - \bar{y} \rangle = \|x^k - \bar{x}\|^2 - \|x^k - \bar{y}\|^2 + \|\bar{x}\|^2 - \|\bar{y}\|^2,$$

whose right-hand side converges to the constant $c' := c_x^2 - c_y^2 + \|\bar{x}\|^2 - \|\bar{y}\|^2$ as $k \rightarrow \infty$. Passing the limits of (3.12) over the two subsequences that weakly converge to \bar{x} and to \bar{y} , respectively, yields $2\langle \bar{x}, \bar{x} - \bar{y} \rangle = 2\langle \bar{y}, \bar{x} - \bar{y} \rangle = c'$. Hence, $\|\bar{x} - \bar{y}\|^2 = 0$ and $(x^k)_{k \geq 0}$ weakly converges to $\bar{x} \in \text{zer}(S)$. \square

It follows immediately from [11, Lemma 3] that

$$\min_{j \leq k} \left\{ \frac{1}{\sqrt{j}} \|Sx^j\|^2 \right\} = o(1/k).$$

Therefore we have the following Corollary.

COROLLARY 3.4. *Under the setting of Theorem 3.3, the reduction rate of running-*

minimal residual is

$$\min_{j \leq k} \{\|Sx^j\|^2\} = o(1/\sqrt{k}).$$

Note that we do not write $\min_{j \leq k} \{\|Sx^j\|\} = o(1/k^{1/4})$ since $\|Sx^k\|^2$ naturally appears in our analysis.

In general we do not expect to have a convergence rate in $\|x^k - x^*\|$ without further assumptions.

3.2. Fixed step size. In Theorem 3.3, α_k in (3.9) are decreasing. Next, we study convergence under the fixed step size $\alpha_k \equiv \alpha < 1$. We need an additional assumption as follows:

ASSUMPTION 2 (quasi- μ -strong monotonicity). *There exists some $\mu > 0$ such that the operator S satisfies*

$$(3.13) \quad \langle Sx, x - x^* \rangle \geq \mu \|x - x^*\|^2, \quad \forall x^* \in \text{zer}(S), x \in \mathcal{H}.$$

This assumption ensures that x^* is the only element in $\text{zer}(S)$. Indeed, any $\bar{x} \in \text{zer}(S)$ must obey $0 = \langle S\bar{x}, \bar{x} - x^* \rangle \geq \mu^2 \|\bar{x} - x^*\|^2$.

Assumption 2 also implies $\|Sx\| \geq \mu \|x - x^*\|$. Note that $\|Sx\| = \|Sx - Sx^*\| \leq 2\|x - x^*\|$, so $\mu \leq 2$ is known.

THEOREM 3.5. *Under Assumptions 1 and 2, using the fixed step size*

$$(3.14) \quad \alpha = \min \left\{ \frac{1}{4mL}, \frac{\mu}{4\sqrt{2}mL}, \frac{2mL}{17mL + 2\mu^2} \right\},$$

Algorithm 1 generates a sequence x^k that converges to $x^* \in \text{zer}(S)$. We further have

$$\|x^k - x^*\|^2 \leq \rho^k \|x^0 - x^*\|^2$$

with $\rho = 1 - \frac{\alpha\mu^2}{2}$. When m and $\frac{L}{\mu}$ are large, we have $\alpha = O(\frac{\mu}{mL})$.

Proof. Combining (2.7) and (3.5) with $x = x^k$ and using $x^{k+1} = E^\alpha x^k$, we obtain

$$\|x^{k+1} - x^*\|^2 \leq (1 + \alpha\eta) \left(\|x^k - x^*\|^2 - \left(\alpha(1 - \alpha) - \eta^{-1}\alpha^3 L^2 \frac{m^2(1+\alpha L)^{2m}}{2} \right) \|Sx^k\|^2 \right).$$

We use (3.13) to get that

$$\begin{aligned} \|x^{k+1} - x^*\|^2 &\leq \|x^k - x^*\|^2 + \frac{\alpha\eta}{\mu^2} \|Sx^k\|^2 \\ &\quad - (1 + \alpha\eta) \left(\alpha(1 - \alpha) - \eta^{-1}\alpha^3 L^2 \frac{m^2(1+\alpha L)^{2m}}{2} \right) \|Sx^k\|^2. \end{aligned}$$

As long as we can ensure

$$(3.15) \quad \frac{1}{2} + \frac{\eta}{\mu^2} - (1 + \alpha\eta) \left(1 - \alpha - \eta^{-1}\alpha^2 L^2 \frac{m^2(1+\alpha L)^{2m}}{2} \right) \leq 0,$$

we would have

$$\|x^{k+1} - x^*\|^2 \leq \|x^k - x^*\|^2 - \frac{\alpha}{2} \|Sx^k\|^2 \leq \left(1 - \frac{\alpha\mu^2}{2}\right) \|x^k - x^*\|^2.$$

Introduce β to rewrite $\alpha = \frac{\beta}{2mL}$. Let $\eta = \frac{\mu^2}{4}$. By $\left(1 + \frac{\beta}{2m}\right)^{2m} < e^\beta$, we have (3.15) if

$$(3.16) \quad \frac{3}{4} - \left(1 + \frac{\beta\mu^2}{8mL}\right) \left(1 - \frac{\beta}{2mL} - \frac{\beta^2 e^\beta}{2\mu^2}\right) \leq 0.$$

By simplification, (3.16) is equivalent to

$$\frac{\beta^3 e^\beta}{16mL} + \beta^2 \left(\frac{e^\beta}{2\mu^2} + \frac{\mu^2}{8m^2 L^2}\right) + \beta \left(\frac{1}{2mL} - \frac{\mu^2}{8mL}\right) \leq \frac{1}{4},$$

When $\beta \leq \frac{1}{2}$, we have $e^\beta < 2$ and $e^\beta \beta < 1$. Hence, we only need to ensure

$$\frac{\beta}{32mL} + \left(\frac{\beta^2}{\mu^2} + \frac{\mu^2 \beta}{16m^2 L^2}\right) + \beta \left(\frac{1}{2mL} - \frac{\mu^2}{8mL}\right) \leq \frac{1}{4},$$

which can be guaranteed by

$$\frac{\beta^2}{\mu^2} \leq \frac{1}{8} \text{ and } \beta \left(\frac{1}{32mL} + \frac{1}{2mL} + \frac{\mu^2}{16m^2 L^2}\right) \leq \frac{1}{8}.$$

Therefore we need

$$\beta \leq \frac{\mu}{2\sqrt{2}} \text{ and } \beta \leq \frac{1}{\frac{17}{4mL} + \frac{\mu^2}{2m^2 L^2}}.$$

Therefore, we arrive at (3.14). □

3.3. Primal-dual coordinate update and its convergence metric. In this section we briefly review primal-dual algorithms and explain how to adapt Algorithm 1 and its analysis in Subsections 3.1 and 3.2.

Primal-dual algorithms [5, 8, 37] can solve the following problem:

$$\underset{x \in \mathcal{H}}{\text{minimize}} \quad g(x) + h(x) + f(Ax),$$

where g is a differentiable convex function; $f, h : \mathcal{H} \rightarrow \mathbb{R} \cup \{\infty\}$ are extended-value convex functions, which are not necessarily differentiable, and $A : \mathcal{H} \rightarrow \mathcal{G}$ is a linear operator from \mathcal{H} to another Hilbert space \mathcal{G} . Through indicator functions, f, h can model constraints like $x \in \mathcal{C}$ or $Ax \in \mathcal{C}$, where \mathcal{C} is some closed convex set. Primal-dual algorithms involve a dual variable $s \in \mathcal{G}$ and iteratively update both x and s by the iteration:

$$(3.17) \quad \begin{cases} x^{k+1} = \mathbf{prox}_{\eta h}(x^k - \eta(\nabla g(x^k) + A^\top s^k)), \\ s^{k+1} = \mathbf{prox}_{\gamma f^*}(s^k + \gamma A(2x^{k+1} - x^k)), \end{cases}$$

where f^* is the Fenchel dual of f and $\mathbf{prox}_{\eta h}$ is the proximal operator of the function h

defined as

$$\mathbf{prox}_{\eta h}(x) := \arg \min_{y \in \mathcal{H}} h(y) + \frac{1}{2\eta} \|y - x\|^2.$$

Define $z := \begin{pmatrix} x \\ s \end{pmatrix}$ and rewrite (3.17) as $z^{k+1} = Tz^k$. It is shown [6] that (with proper choice of step sizes) T is nonexpansive under the metric induced by the norm $\|z\|_M = \sqrt{\langle z, Mz \rangle}$, where M is a certain positive definite linear operator.

To describe our coordinate-update algorithm, we assume that a product-form Hilbert space $\mathcal{G} = \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \times \mathcal{G}_p$ and break the dual variable $s \in \mathcal{G}$ into p blocks:

$$s = (s_1, s_2, \dots, s_p)$$

with $s_i \in \mathcal{G}_i$, $i = 1, 2, \dots, p$. Each step of Algorithm 1 picks a coordinate z_i of z , which can be a coordinate of either x or s , and follows (3.17) to update z_i only. We use the techniques in [28, Section 4] to ensure such coordinate updates computationally worthy. That is, updating one coordinate of x or s only takes $O(\frac{1}{m+p})$ of the cost of computing the full update (3.17).

When $g = 0$ and $h = 0$ (which is the case of the test problems presented in Section 4), $\mathbf{prox}_{\eta h}$ is the identity operator and, thus, we can plug the x update into the s update and obtain the simpler iteration:

$$(3.18) \quad \begin{cases} x^{k+1} = x^k - \eta A^\top s^k, \\ s^{k+1} = \mathbf{prox}_{\gamma f^*}(s^k + \gamma Ax^k - 2\gamma \eta AA^\top s^k), \end{cases}$$

where s^{k+1} can be computed from x^k and s^k . In contrast, computing s^{k+1} in (3.17) requires x^{k+1} .

Computing coordinate updates requires the *cache-then-update* technique. In particular, for the coordinate update based on (3.18), we cache the variable (Ax^k) when $m \gg p$. To compute s_i^{k+1} , we directly use $(Ax^k)_i$ instead of multiplying $A_{i,:}$ and x^k . When x_i^k is updated to x_i^{k+1} , we update (Ax^k) to (Ax^{k+1}) following $Ax^{k+1} = Ax^k + A_{:,i} \cdot (x_i^{k+1} - x_i^k)$, which takes only $O(p)$ operations. This is cheaper than computing Ax directly, which takes $O(m)$ numbers of operations.

Our proofs in Sections 3.1 and 3.2 apply to primal-dual coordinate-update algorithms after adjusting certain constants for $\|\cdot\|_M$ by the next lemma.

LEMMA 3.6. *Let $\lambda_{\max}, \lambda_{\min}$ be the largest and smallest eigenvalues of M , respectively, and $\kappa := \frac{\lambda_{\max}}{\lambda_{\min}}$ be its condition number. We have for any $z \in \mathcal{H}$,*

$$\frac{1}{\kappa^2} \|z\|_M^2 \leq \sum_{i=1}^m \|z_i\|_M^2 \leq \kappa^2 \|z\|_M^2.$$

Proof.

$$\sum_{i=1}^m \|z_i\|_M^2 \leq \sum_{i=1}^m \lambda_{\max}^2 \|z_i\|^2 = \lambda_{\max}^2 \|z\|^2 \leq \frac{\lambda_{\max}^2}{\lambda_{\min}^2} \|z\|_M^2 = \kappa^2 \|z\|_M^2.$$

The other half is similar. \square

In particular, when S is $1/2$ -cocoercive under the norm $\|\cdot\|_M$, we have $\|S_i x - S_i y\|_M \leq 2\kappa \|x - y\|_M$ for all i . Without a proof, we state an extension to Lemma 3.1.

LEMMA 3.7. *Let M be a symmetric positive definite matrix with condition number κ , $I - S$ be nonexpansive under the norm $\|\cdot\|_M$, T^α , E^α and R be defined as in (2.4), (2.5) and (3.1), and L be defined as (2.3). The operator R satisfies the estimate*

$$\|Rx\| \leq \frac{\alpha L m \kappa^2}{\sqrt{2}} (1 + \alpha L)^m \|Sx\|.$$

Based on Lemma 3.7, the statement of Theorem 3.3 still holds, and the proof is similar except that the choice of η is slightly adjusted. Theorem 3.5 still holds, too, but with the step size $\alpha = O(\frac{\mu}{\kappa m L})$ and some new constants in its proof.

4. Numerical experiments. In this section we illustrate the efficiency of Algorithm 1 on three different applications: ℓ_1 based robust linear regression, computed tomography and nonnegative matrix factorization. They are important problems in statistics, medical imaging, and machine learning, respectively. The first two problems cannot be solved by the traditional coordinate descent algorithms, and the last one is a nonconvex problem.

We use these results to illustrate the following two points:

- In spite of the small theoretical step sizes, practical problems in our preliminary experiments accept very large step sizes, which contribute to the great performance of our Algorithm 1.
- Our Algorithm 1 is significantly faster than the standard fixed-point iteration, which performs the full update in each iteration, and also faster than the algorithm using randomized coordinate selection.

In all of our numerical experiments, convergence was observed with $\alpha_k = 1$ in Algorithm 1. In addition, Algorithm 1 and its randomized variant both admit larger *intrinsic* step sizes, which are η, γ in (3.18). This brings a significant speed advantage to the coordinate update algorithms over the standard fixed-point iteration.

Our numerical experiments are coded in Matlab that is running on a laptop with 2.7 GHz Intel Core i5 and 8 Gigabytes of RAM.

4.1. ℓ_1 based robust linear regression. Consider the problem:

$$(4.1) \quad \underset{x \in \mathbb{R}^m}{\text{minimize}} \quad f(Ax) := \|Ax - b\|_1,$$

where $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$ are given. We apply Algorithm 1 to the Chambolle-Pock operator with diagonal scaling [31] to solve (4.1). Specifically, the fixed-point iteration is:

$$(4.2a) \quad x^{k+1} = x^k - HA^\top s^k,$$

$$(4.2b) \quad s^{k+1} = \mathbf{prox}_{\Gamma f^*}(s^k + \Gamma A(2x^{k+1} - x^k)),$$

where $f^*(y) = \iota_{\|\cdot\|_\infty \leq 1}(y) + y^\top b$ ³ and H, Γ are diagonally scaling matrices with $H_{ii} = \frac{1}{\|A_{:,i}\|_1}$, $\Gamma_{ii} = \frac{1}{\|A_{i,:}\|_1}$. Furthermore, we have

$$(4.3) \quad \mathbf{prox}_{\Gamma f^*}(y) := \arg \min_t f^*(t) + \frac{1}{2} \|t - y\|_{\Gamma^{-1}}^2 = \text{Proj}_{\|\cdot\|_\infty \leq 1}(y - \Gamma b).$$

Here, $y = \text{Proj}_{\|\cdot\|_\infty \leq 1}(x)$ can be computed component-wise as $y_i = \text{Proj}_{[-1,1]}(x_i) = \max\{-1, \min\{1, x_i\}\}$, $i = 1, \dots, n$. Substituting the x update (4.2a) into the s update (4.2b) and using (4.3), we can rewrite (4.2) as

$$(4.4a) \quad x^{k+1} = x^k - HA^\top s^k,$$

$$(4.4b) \quad s^{k+1} = \text{Proj}_{\|\cdot\|_\infty \leq 1}(s^k - \Gamma b + \Gamma A(x^k - 2HA^\top s^k)),$$

which is more suitable for coordinate update.

We define $z := \begin{pmatrix} x \\ s \end{pmatrix}$ and write (4.4) as $z^{k+1} := Tz^k$. To this operator T and $S = I - T$ do we apply Algorithm 1.

In our experiments, we let $n = 500$ and $m = 100$. The elements of A and b are sampled from the standard normal distribution. The solution x^* and the optimal function value f^* are obtained by solving Problem (4.1) using CVX. The starting point $z^0 = \begin{pmatrix} x^0 \\ s^0 \end{pmatrix}$ is set to be 0. The block size is 1.

Since Problem (4.1) may have multiple solutions, we use $\frac{f^k - f^*}{f^*}$ to measure convergence. To fully explore the power of each algorithm, we multiply both step size matrices Γ and H in (4.4) by a scaling factor ν . We set $\nu = 6$ for the full update (a larger value leads to divergence) and $\nu = 12$ for the coordinate updates. Figure 1 plots $\frac{f^k - f^*}{f^*}$ versus epoch. We can see that the cyclic and shuffled cyclic algorithms perform better than the random algorithm, which is further faster than the full update algorithm.

4.2. Computed tomography (CT). Consider the image recovery problem:

$$(4.5) \quad \underset{x \in \mathbb{R}^m}{\text{minimize}} \quad \lambda \|\nabla x\|_1 + \frac{1}{2} \|Ax - b\|^2,$$

where $x \in \mathbb{R}^m$ is the unknown two-dimensional image (reformulated as a vector), $\nabla \in \mathbb{R}^{n_1 \times m}$ is the finite difference operator, λ is a scaling factor, $A \in \mathbb{R}^{n_2 \times m}$ is the Radon transform matrix, and $b \in \mathbb{R}^{n_2}$ is the observed CT data, which is contaminated by random noise. Let $n := n_1 + n_2$.

³ ι_C is the indicator function: $\iota_C(x) = 0$ if $x \in C$ and $= \infty$ if $x \notin C$.

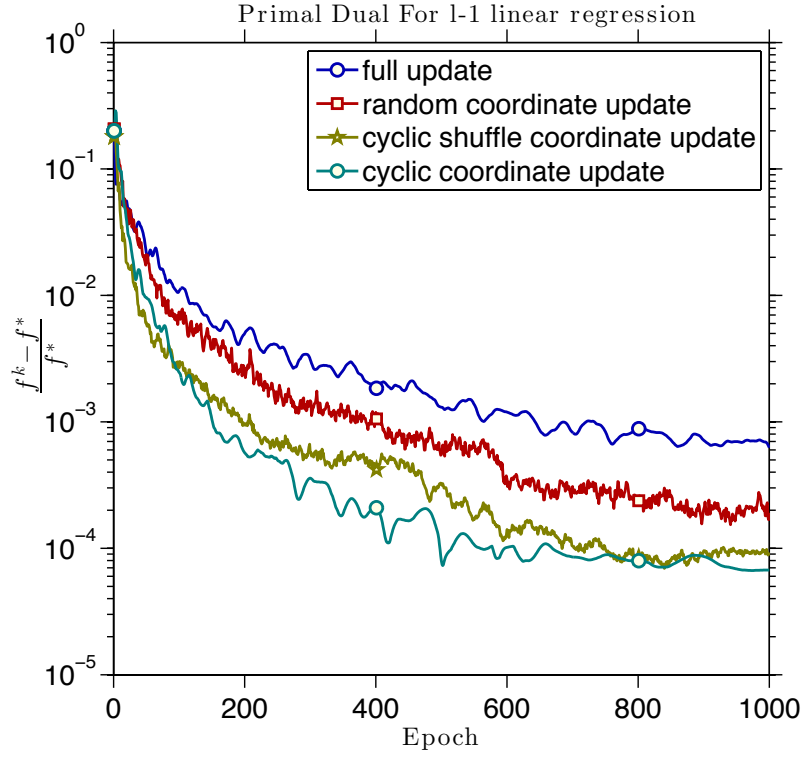


Figure 1: ℓ_1 based robust least squares

By defining

$$B := \begin{pmatrix} \nabla \\ A \end{pmatrix}, \quad f(p, q) := \lambda \|p\|_1 + \frac{1}{2} \|q - b\|^2, \text{ for } p \in \mathbb{R}^{n_1}, q \in \mathbb{R}^{n_2},$$

we can rewrite (4.5) as

$$\underset{x \in \mathbb{R}^m}{\text{minimize}} f(Bx).$$

To solve this problem, we apply Algorithm 1 to the fixed-point iteration (see [28, section 5.2.2] for its derivation):

$$\begin{aligned} x^{k+1} &= x^k - \eta(\nabla^\top s^k + A^\top t^k), \\ s^{k+1} &= \text{Proj}_{\|\cdot\|_\infty \leq \lambda} (s^k + \gamma \nabla(x^k - 2\eta(\nabla^\top s^k + A^\top t^k))), \\ t^{k+1} &= \frac{1}{1 + \gamma} (t^k + \gamma A(x^k - 2\eta(\nabla^\top s^k + A^\top t^k)) - \gamma b). \end{aligned}$$

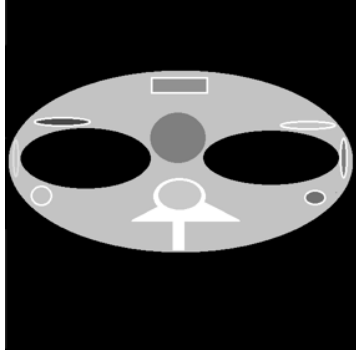
We implement Algorithm 1 with $\alpha_k = 1$ and compare it to the full update and random coordinate selection.

We generate a thorax phantom of size 284×284 . The Radon matrix A is generated by

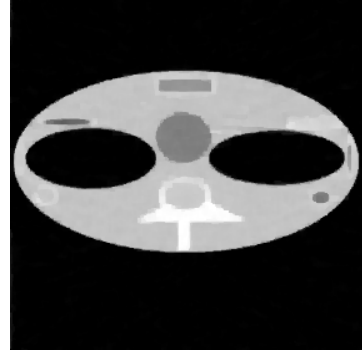
Siddon's algorithm [35].

The image x is partitioned into 284 blocks, with each block corresponding to a column of the image. The dual variables s, t are also partitioned into 284 blocks accordingly. A block of x and the corresponding blocks of s and t are bundled together as a single block. In each iteration, a bundled block of x, s, t is chosen and updated.

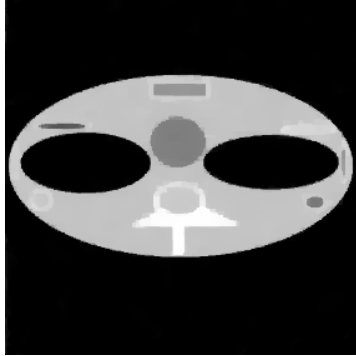
The step sizes η and γ are hand tuned for both the full and coordinate updates. The different rules of coordinate selection use the same step size.



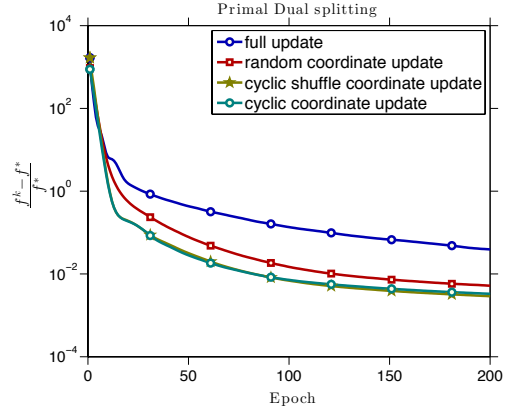
(a) Phantom image



(b) Recovered by full update



(c) Recovered by cyclic coordinate update



(d) Objective function value

Figure 2: CT image reconstruction.

Figure 2 depicts the results. After 200 epochs, the cyclic algorithm recovers the original image (Figure 2a) much better than the full update algorithm. Compare Figures 2c and 2b. We use the TVAL3 package⁴ [21] to obtain a high-accuracy objective value f^* of (4.5) and plot $\frac{f^k - f^*}{f^*}$ versus epoch in Figure 2d. We can see that the cyclic or cyclic shuffle versions

⁴Accessed from <http://www.caam.rice.edu/~optimization/L1/TVAL3/> on Oct. 26, 2016.

of Algorithm 1 have similar performance, and they are faster than random selection. All the three coordinate algorithms are faster than the full update algorithm.

4.3. Nonnegative matrix factorization. Consider the problem:

$$\underset{X \in \mathbb{R}_+^{n \times r}, Y \in \mathbb{R}_+^{m \times r}}{\text{minimize}} \quad \frac{1}{2} \|XY^\top - M\|_F^2,$$

where $0 < r \ll \min(m, n)$ and $M \in \mathbb{R}_+^{n \times m}$ are given. This problem is nonconvex. Since the objective function is biconvex (convex in X while Y is fixed, and vice versa), we can apply the alternating projected gradient iteration, which we treat as a fixed-point iteration $(X^{k+1}, Y^{k+1}) = T(X^k, Y^k)$:

$$(4.7) \quad \begin{cases} X^{k+1} = \max(0, X^k - \alpha_k \nabla_X f(X^k, Y^k)), \\ Y^{k+1} = \max(0, Y^k - \beta_k \nabla_Y f(X^{k+1}, Y^k), \end{cases}$$

where $\alpha_k, \beta_k > 0$ are step sizes and

$$\begin{cases} \nabla_X f(X, Y) = (XY^\top - M)Y, \\ \nabla_Y f(X, Y) = (YX^\top - M^\top)X. \end{cases}$$

We partition X, Y into columns $X = [X_1 \ \cdots \ X_r]$ and $Y = [Y_1 \ \cdots \ Y_r]$ and apply the following coordinate update to (4.7) (notation: $X_{<i} := [X_1 \ \cdots \ X_{i-1}]$):

$$(4.8) \quad \begin{cases} X_i^{k+1} = \arg \min_{X_i \geq 0} \frac{1}{2} \|X_i(Y_i^k)^\top + X_{<i}^{k+1}(Y_{<i}^{k+1})^\top + X_{>i}^k(Y_{>i}^k)^\top - M\|_F^2, \\ Y_i^{k+1} = \arg \min_{Y_i \geq 0} \frac{1}{2} \|X_i^{k+1}Y_i^\top + X_{<i}^{k+1}(Y_{<i}^{k+1})^\top + X_{>i}^k(Y_{>i}^k)^\top - M\|_F^2, \end{cases}$$

which appears in the recent work [18, 42].

Each problem in (4.8) has closed form solutions as follows:

$$(4.9) \quad \begin{cases} X_i^{k+1} = \text{Proj}_{\mathbb{R}_+^n} \left[X_i^k - \frac{1}{L_{X_i}^k} \nabla_{X_i} f(X_{<i}^{k+1}, X_{\geq i}^k, Y_{<i}^{k+1}, Y_{\geq i}^k) \right], \\ Y_i^{k+1} = \text{Proj}_{\mathbb{R}_+^m} \left[Y_i^k - \frac{1}{L_{Y_i}^k} \nabla_{Y_i} f(X_{\leq i}^{k+1}, X_{>i}^k, Y_{<i}^{k+1}, Y_{\geq i}^k) \right], \end{cases}$$

where $\nabla_{X_i} f(X, Y) = (XY^\top - M)Y_i$, $\nabla_{Y_i} f(X, Y) = (YX^\top - M^\top)X_i$, and $L_{X_i}^k = \|Y_i^k\|_2^2$, $L_{Y_i}^k = \|X_i^{k+1}\|_2^2$ are their corresponding Lipschitz constants.

An issue of (4.9) is that X_i^{k+1}, Y_i^{k+1} can potentially equal zero. Hence, we make two modifications. Firstly, we force each column of X to have unit length (notice $XY^\top = (XU)(YU^{-1})^\top$ for any matrix $U \in \mathbb{R}_+^{r \times r}$); secondly, we redefine $L_{X_i}^k$ as $L_{X_i}^k = \max(L_{\min}, \|Y_i^k\|_2^2)$. Consequently (4.9) is modified to the following:

$$\begin{cases} X_i^{k+1} = \text{Proj}_{\mathbb{R}_+^n \cap S^{n-1}} \left[X_i^k - \frac{1}{\min(L_{\min}, \|Y_i^k\|_2^2)} \nabla_{X_i} f(X_{<i}^{k+1}, X_{\geq i}^k, Y_{<i}^{k+1}, Y_{\geq i}^k) \right], \\ Y_i^{k+1} = \text{Proj}_{\mathbb{R}_+^m} \left[Y_i^k - \nabla_{Y_i} f(X_{\leq i}^{k+1}, X_{>i}^k, Y_{<i}^{k+1}, Y_{\geq i}^k) \right], \end{cases}$$

which can still be written in the closed form; see [42, Appendix B].

We implement the above coordinate update with the random, cyclic and cyclic shuffle coordinate selection rules and compare their performance with (4.7).

In our experiments, we set $n = 400, m = 400, q = 20$ and generate $M = LR + N_o$, where the elements of L and R are sampled from the standard normal distribution then thresholded positively. The random noise $N_o \in \mathbb{R}^{n \times m}$ is generated in the same way and scaled such that $\|N_o\|_F = 10^{-3}\|LR\|_F$. The constant L_{\min} is set to 0.001. The step sizes in (4.7) are set as $\alpha_k = \frac{1}{\|(Y^k)^\top Y^k\|_2}, \beta_k = \frac{1}{\|(X^{k+1})^\top X^{k+1}\|_2}$.

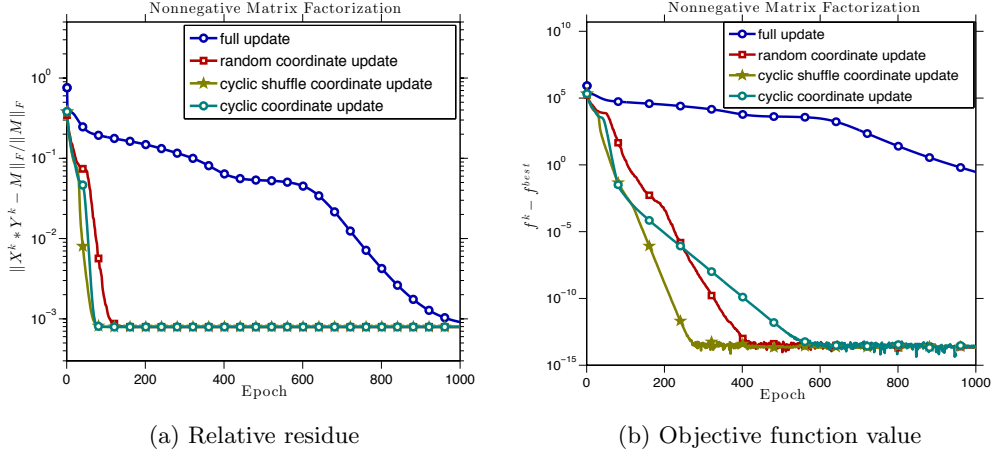


Figure 3: Nonnegative matrix factorization

Figure 3a plots the relative residue $\frac{\|X^k Y^k - M\|_F}{\|M\|_F}$ versus epoch. To better compare the algorithms, we record the smallest function value f^{best} achieved by running all algorithms for 4000 epochs and plot $f^k - f^{\text{best}}$ versus epoch in Figure 3b. The comparison results are similar, except that cyclic update becomes slightly slower than the other two after reaching a medium accuracy.

5. Conclusion. We have proposed a cyclic coordinate-update algorithm for the non-expansive fixed-point problem, established its convergence with slowly decreasing step sizes and, under a stronger condition, with a fixed step size. Numerical results illustrate the higher efficiency of the proposed algorithms over the traditional fixed-point iteration. Also, cyclic selection is overall more efficient than random selection.

Acknowledgements. We thank Robert Hannah, Zhimin Peng, Yangyang Xu, and Ming Yan for motivating discussions, as well as Qing Ling for his comments. We also thank Zhimin Peng for sharing his CT code.

REFERENCES

- [1] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books in Mathematics, Springer New York, New York, NY, 2011.
- [2] A. BECK AND L. TETRUASHVILI, *On the convergence of block coordinate descent type methods*, SIAM Journal on Optimization, 23 (2013), pp. 2037–2060.
- [3] D. P. BERTSEKAS, *Distributed asynchronous computation of fixed points*, Mathematical Programming, 27 (1983), pp. 107–120.
- [4] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and distributed computation: numerical methods*, Prentice hall Englewood Cliffs, NJ, 1989.
- [5] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.
- [6] P. L. COMBETTES, L. CONDAT, J.-C. PESQUET, AND B. C. VU, *A forward-backward view of some primal-dual optimization methods in image recovery*, in Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 4141–4145.
- [7] P. L. COMBETTES AND J.-C. PESQUET, *Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping*, SIAM Journal on Optimization, 25 (2015), pp. 1221–1248.
- [8] L. CONDAT, *A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms*, Journal of Optimization Theory and Applications, 158 (2013), pp. 460–479.
- [9] A. P. DA COSTA AND A. SEEGER, *Cone-constrained eigenvalue problems: theory and algorithms*, Computational Optimization and Applications, 45 (2010), pp. 25–57.
- [10] D. DAVIS AND W. YIN, *A three-operator splitting scheme and its optimization applications*, UCLA CAM Report 15-13, (2015).
- [11] D. DAVIS AND W. YIN, *Convergence rate analysis of several splitting schemes*, in Splitting Methods in Communication, Imaging, Science and Engineering, R. Glowinski, S. Osher, and W. Yin, eds., Chapter 4, Springer, 2016.
- [12] O. FERCOQ AND P. BIANCHI, *A coordinate descent primal-dual algorithm with large step size and possibly non separable functions*, arXiv preprint arXiv:1508.04625, (2015).
- [13] J. FRIEDMAN, T. HASTIE, H. HÖFLING, R. TIBSHIRANI, ET AL., *Pathwise coordinate optimization*, The Annals of Applied Statistics, 1 (2007), pp. 302–332.
- [14] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI, *Regularization paths for generalized linear models via coordinate descent*, Journal of statistical software, 33 (2010), p. 1.
- [15] D. GABAY AND B. MERCIER, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.
- [16] R. GLOWINSKI AND A. MARROCO, *Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires*, ESAIM: Mathematical Modelling and Numerical Analysis, 9 (1975), pp. 41–76.
- [17] M. HANKE, A. NEUBAUER, AND O. SCHERZER, *A convergence analysis of the landweber iteration for nonlinear ill-posed problems*, Numerische Mathematik, 72 (1995), pp. 21–37.
- [18] N.-D. HO, P. VAN DOOREN, AND V. D. BLONDEL, *Descent methods for nonnegative matrix factorization*, in Numerical Linear Algebra in Signals, Systems and Control, Springer, 2011, pp. 251–293.
- [19] A. N. IUSEM AND W. SOSA, *On the proximal point method for equilibrium problems in hilbert spaces*, Optimization, 59 (2010), pp. 1259–1274.
- [20] M. KRASNOSEL’SII, *Two remarks on the method of successive approximations*, Uspekhi Matematicheskikh Nauk, 10 (1955), pp. 123–127.
- [21] C. LI, W. YIN, H. JIANG, AND Y. ZHANG, *An efficient augmented Lagrangian method with applications to total variation minimization*, Computational Optimization and Applications, 56 (2013), pp. 507–530.
- [22] P. L. LIONS AND B. MERCIER, *Splitting Algorithms for the Sum of Two Nonlinear Operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964–979.
- [23] Z. LU AND L. XIAO, *On the complexity analysis of randomized block-coordinate descent methods*, Mathematical Programming, 152 (2015), pp. 615–642.
- [24] Z.-Q. LUO AND P. TSENG, *On the convergence of the coordinate descent method for convex differen-*

- table minimization*, Journal of Optimization Theory and Applications, 72 (1992), pp. 7–35.
- [25] W. R. MANN, *Mean value methods in iteration*, Proceedings of the American Mathematical Society, 4 (1953), pp. 506–510.
 - [26] Y. NESTEROV, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362.
 - [27] R. NONG AND D. C. SORENSEN, *A parameter free ADI-like method for the numerical solution of large scale lyapunov equations*, Computational and Applied Mathematics, (2009).
 - [28] Z. PENG, T. WU, Y. XU, M. YAN, AND W. YIN, *Coordinate friendly structures, algorithms and applications*, Annals of Mathematical Sciences and Applications, 1 (2016), pp. 57–119.
 - [29] Z. PENG, Y. XU, M. YAN, AND W. YIN, *ARock: an algorithmic framework for asynchronous parallel coordinate updates*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2851–A2879.
 - [30] J.-C. PESQUET AND A. REPETTI, *A class of randomized primal-dual algorithms for distributed optimization*, Journal of Nonlinear and Convex Analysis, 16 (2015), pp. 2453–2490.
 - [31] T. POCK AND A. CHAMBOLLE, *Diagonal preconditioning for first order primal-dual algorithms in convex optimization*, in 2011 International Conference on Computer Vision, IEEE, 2011, pp. 1762–1769.
 - [32] M. RAZAVIYAYN, M. HONG, AND Z.-Q. LUO, *A unified convergence analysis of block successive minimization methods for nonsmooth optimization*, SIAM Journal on Optimization, 23 (2013), pp. 1126–1153.
 - [33] P. RICHTÁRIK AND M. TAKÁČ, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming, 144 (2014), pp. 1–38.
 - [34] H.-J. M. SHI, S. TU, Y. XU, AND W. YIN, *A primer on coordinate descent algorithms*, arXiv preprint arXiv:1610.00040, (2016).
 - [35] R. L. SIDDON, *Fast calculation of the exact radiological path for a three-dimensional ct array*, Medical physics, 12 (1985), pp. 252–255.
 - [36] P. TSENG, *Convergence of a block coordinate descent method for nondifferentiable minimization*, Journal of optimization theory and applications, 109 (2001), pp. 475–494.
 - [37] B. C. VŨ, *A splitting algorithm for dual monotone inclusions involving cocoercive operators*, Advances in Computational Mathematics, 38 (2013), pp. 667–681.
 - [38] J. WARGA, *Minimizing certain convex functions*, Journal of the Society for Industrial and Applied Mathematics, 11 (1963), pp. 588–593.
 - [39] S. J. WRIGHT, *Coordinate descent algorithms*, Mathematical Programming, 151 (2015), pp. 3–34.
 - [40] F.-Q. XIA AND N.-J. HUANG, *A projection-proximal point algorithm for solving generalized variational inequalities*, Journal of optimization theory and applications, 150 (2011), pp. 98–117.
 - [41] Y. XU AND W. YIN, *A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion*, SIAM Journal on imaging sciences, 6 (2013), pp. 1758–1789.
 - [42] Y. XU AND W. YIN, *A globally convergent algorithm for nonconvex optimization based on block coordinate update*, arXiv preprint arXiv:1410.1386. To appear in Journal of Scientific Computing, (2014).
 - [43] G.-X. YUAN, K.-W. CHANG, C.-J. HSIEH, AND C.-J. LIN, *A comparison of optimization methods and software for large-scale l1-regularized linear classification*, Journal of Machine Learning Research, 11 (2010), pp. 3183–3234.